***Blogic,* a web-based introduction to symbolic logic**

J. David Velleman
New York University (jdvelleman@nyu.edu)
Johns Hopkins University (dvellem_1@jh.edu)

Blogic is a web-based, interactive course in symbolic logic covering the standard syllabus taught by most U.S. philosophy departments in a first logic course, from propositional logic through quantified predicate logic, with natural-deduction proofs in both. It consists of a textbook with interactive manipulables that simulate logical concepts in various ways. Those manipulables illustrate the text and are incorporated into ungraded exercises. A coordinated set of graded quizzes and course-management tools can be hosted together with the text in a php/mysql environment.

Design of the Course

The course is designed to circumvent the difficulties that many intelligent undergraduates experience in logic, especially if they are not mathematically inclined. The course introduces the same symbolic systems of a standard logic course, but it makes them more intelligible to students for whom abstract symbols are not second-nature.

The exercises in each chapter take full advantage of the interactive features that illustrate the exposition of the text. When they ask true/false or multiple-choice questions, they require the student to correct his or her errors using interactive logic circuits, Venn diagrams, truth tables, possible-worlds diagrams, and so on. This feedback and opportunity for correction is provided on the spot, while the student still remembers the thinking behind his or her initial response.

In the credit-bearing course, students take one or two computer-scored quizzes per week. Grading is based on a “mastery” system. Instead of starting with 100 points and losing points for errors, students start with 0 points and earn points for successfully completing an item. Because the software helps them to correct their initial errors, they can all reach completion, but the more errors they make along the way, the fewer points they earn.

Quizzes are unscheduled, untimed, and can be interrupted at any point and resumed later. If students find themselves making too many errors, they have an incentive to interrupt their work, review the text, practice with the ungraded exercises, or ask for help,. The quizzes thus enable them to diagnose and seek the means to correct their own misunderstandings. The goal of this design — explicitly stated at the outset of the course — is for every student to earn an A. Students are thus encouraged to view logic as a skill that they can master, not a test of their intelligence.

Chapter Summary

This approach begins in **Chapter 1**, which introduces the Boolean operators in the context of database searching in natural language. All undergraduates have searched not only Google or Amazon but also their library's online catalog, where Boolean connectives are prominently offered even if rarely used. Students are therefore familiar with idea of combining simple descriptors with 'and', 'or', and 'not' to select items from a collection. This idea can later be applied, not only in the context of predicate logic, but also in propositional logic, where compound statements select possible worlds from the universe of possibilities. But there is no reason to introduce symbols at this stage, because the natural-language connectives are less ambiguous in the context of search queries than in the context of ordinary discourse: there is no temptation interpret 'or' as exclusive or 'and' as implicating temporal order in a search query. Natural-language search queries are therefore preferable as an introduction to the connectives.

Boolean searching is also helpful for introducing the concept of informativeness as the ruling out of possibilities: a search is uninformative if it doesn't sufficiently narrow down the results. This concept is used later to introduce conversational implicature, which is helpful in clearing up misreadings of the logical connectives, because they often differ from their ordinary-language homonyms precisely by lacking common implicatures. The course also forestalls misreadings by replacing "if … then", for the material conditional, with "only if", which carries fewer misleading connotations.

The manipulables in Chapter 1 include interactive logic circuits with which students can experiment to see how 'and', 'or', and 'not' gates combine to perform a Boolean search, and interactive Venn diagrams with which they can visualize the extensions of, and logical relations between, compound predicates. The software can produce circuits to implement arbitrary search queries using four keywords, and circuits implementing different queries can be produced side-by-side so that students can compare their output in response to the same input. They can also use Venn diagrams to visualize the logical relations of equivalence, necessity, and sufficiency between compound predicates. The circuits and diagrams help them to acquire, not only an abstract understanding of Boolean logic, but also a "feel" for it by highlighting different regions of a diagram, or by turning switches on and off and seeing how the current flows in different circuits.

The chapter also provides an interactive "wff processor" that applies the recursive rules of Boolean syntax to compose search queries. Students observe that the sequence in which a query is compounded corresponds to the sequence in which a logic circuit is built, which corresponds in turn to the sequence in which current flows through the gates of the circuit.

Chapter 2 begins by introducing the concept of possible worlds, with schematic drawings representing the permutations of truth and falsity in four simple statements. Students can highlight or dim these drawings to see which worlds satisfy Boolean compounds of those statements. The more informative statements require them to dim more "worlds".

This is the point at which the text introduces the symbolic language of propositional logic. The symbols used are more suggestive of their English equivalents than the traditional symbols such as ∨ and ∧. 'Or' is represented by a slash (as in "and/or"), 'and' is represented by an ampersand, and 'if and only if' is represented by a colon, which is sometimes used in to separate equivalent statements.

When students begin to construct truth tables, the permutations of truth and falsity assigned to the variables are enumerated in 4-digit binary notation, with 1 representing truth and 0 representing falsity of the corresponding statement. This scheme serves two purposes. First, it provides a clear explanation of why the distribution of truth values to the variables in a truth table is guaranteed to exhaust the space of possibilities. Every four-digit sequence of 1s and 0s represents a number from 0 to 15 in binary notation, and so we can be sure of including every permutation by counting from 0 to 15 in 4-digit binary. The second purpose of enumerating possibilities by counting is that the initial step in constructing a truth table — namely, distributing truth values among the variables — can be delegated to a clickable binary counter, so that clerical errors at this stage do not defeat a student's efforts later on. Students' can then skip the routine part of the task and devote their attention to the part that tests their understanding. (There is a section in which they are taught how to make the initial distribution of truth values, but it is optional.) They initially construct the tables with the help of logic circuits, again observing that the order in which columns of a table are filled in corresponds to the order in which current travels through the gates of the corresponding circuit.

In this chapter students expand their vocabulary of logical relations to include consistency and inconsistency, contradiction and tautology, and valid inference. They then learn to symbolize and evaluate natural-language arguments, including those with non-standard connectives such as "unless", "provided that", and so on. An appendix presents some of the problems affecting indicative conditionals, drawing on a well-known paper by Robert Stalnaker.

A second appendix to Chapter 2 introduces the concept an effective procedure with a programmable Turing machine. This concept will appear later, when predicate logic is contrasted with propositional logic.

Natural deduction in propositional logic is the topic of **Chapter 3**. Students build derivations by clicking on the statements that serve as the input to a rule and then clicking on the rule, thereby adding the output as the next line in the derivation. They "earn" the use of derived rules by deriving them and causing the clickable rules to appear.

Chapter 4 explains the possible-worlds interpretation of counterfactual conditionals, using possible-worlds diagrams in which schematically represented "worlds" can be moved nearer or further in relation to the actual world, changing the truth-value of arbitrary counterfactuals compounded of four atomic statements. This topic is not part of the traditional logic curriculum, but it provides a relatively easy and (with David Lewis's tipsy kangaroos) entertaining interlude before the introduction of predicate logic.

Chapter 5 introduces predicate logic with a model consisting of colored shapes that can be moved around on a grid. (This model is based on *Tarski's World* by John Barwise and John Etchemendy.) An interactive tool leads students step-by-step through unpacking and evaluating arbitrary formulas as interpreted in application to this model, and exercises provide practice in recognizing the symbolic representations of quantified English statements.

Chapter 6 is devoted to natural deduction in predicate logic, using an enhanced version of derivation-builder of Chapter 4. The system of derivation is adapted from Daniel J. Velleman's "Variable declarations in natural deduction" (*Annals of Pure and Applied Logic* 144 [1]:133-146 [2006]) with the author's assistance.

<u>Access to the Materials</u>

The textbook and exercises are available on the World Wide Web at https://weblogic.a2hosted.com/Logic/

The graded quizzes are available only to students who have enrolled in the course, because each student's progress must be recorded throughout the semester.